

Introduction to Spring Web Flow

Andrew Petro
Software Developer
Unicon, Inc.

Jasig 2011
Westminster, CO
23 May 2011



© Copyright Unicon, Inc., 2011. Some rights reserved. This work is licensed under a Creative Commons Attribution-NonCommercial-Share Alike 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>



Spring Web Flow

1. How Spring Web Flow Fits
2. What's a Flow?
3. Spring Web Flow States
4. Forms

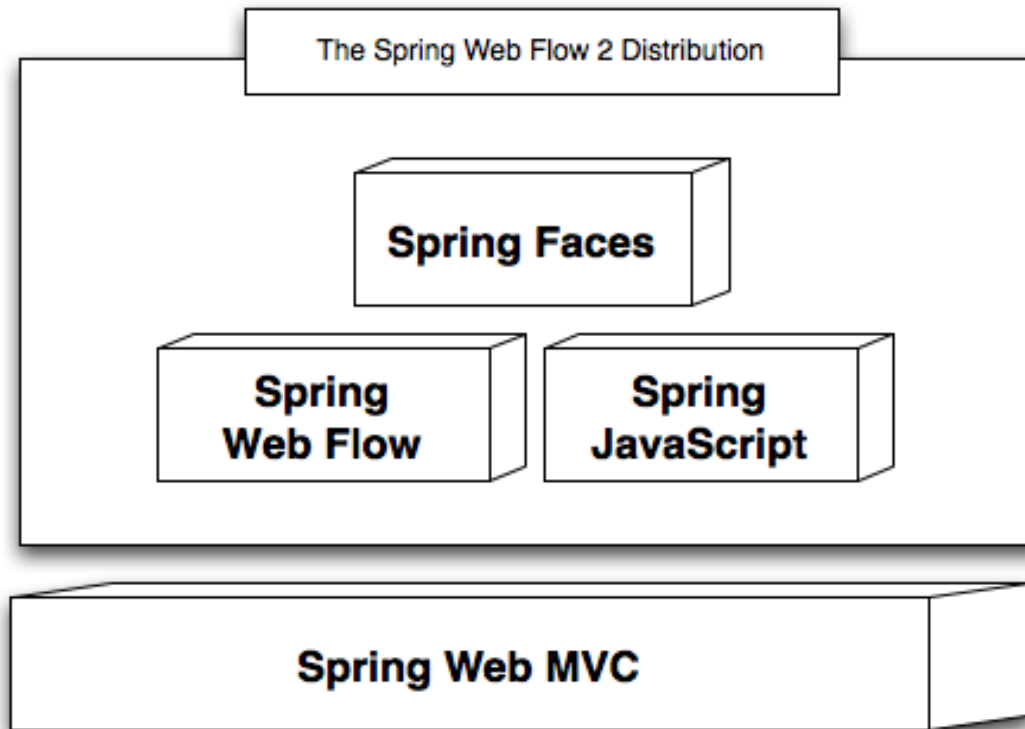
Why is Spring Web Flow important?

- uPortal 4 uses Spring Web Flow
- CAS uses Spring Web Flow
- Spring Web Flow as viable Portlet development framework
- Quickly build and maintain wizard / flow Web experiences with panache
 - Forms, processes, registrations
 - This is bread and butter self-service IT

How Spring Web Flow Fits

Sits atop Spring Framework / MVC

- Spring Web Flow works with the rest of the Spring Framework and Spring WebMVC



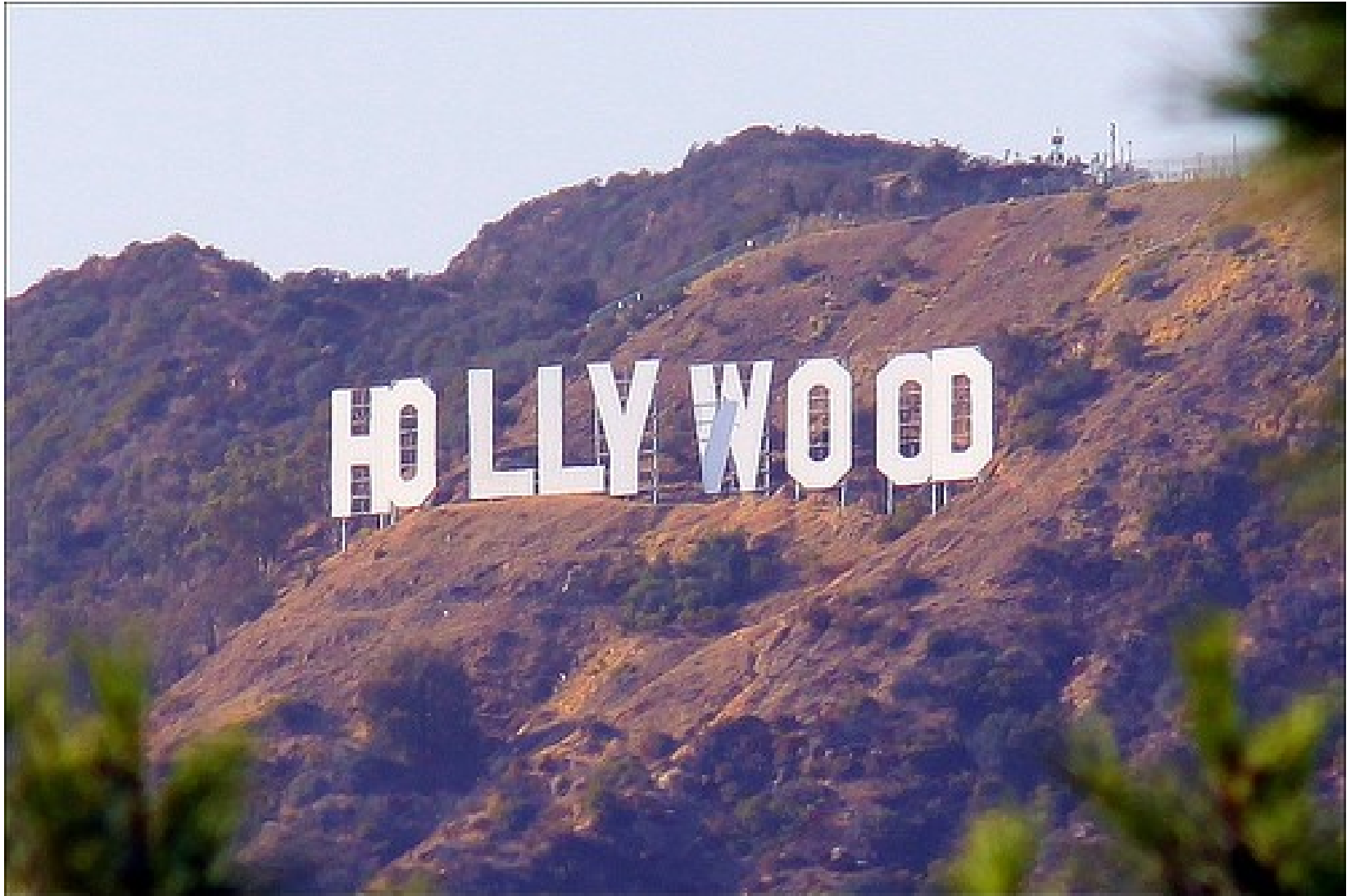
Purpose of Spring Web Flow

- Spring Web Flow is good at building flows
- Which just begs the question, really.
- Let me get at this another way...

Aside: Spring Framework

- Dependency Injection / Inversion of Control
- Utilities, templates, helpers
- For easing all aspects of Java development
 - Especially Java Web development

The Hollywood Principle



Don't Call Us, We'll Call You



Dependency Injection

- Write POJOs. JavaBeans.
- Spring wires together using XML or Annotations or Autowiring or ...
 - Stop calling static factories.
- Code becomes more reusable, more flexibly configurable, less repetitive, ...

Editing Portlet Publications in uPortal 4 RC1

An example of a flow

Portlet Management

PORTLET ADMINISTRATION

[← Return to dashboard](#) [+ Add to my layout](#)

Portlet Registry

[Register New Portlet](#)

[Popular Portlets](#)

Search for stuff

[Search](#)

Show

[< Previous](#) [1](#) [2](#) [3](#) [5](#) [6](#) [7 \(last\)](#) [Next >](#)

Viewing page 1. Showing records 1 - 10 of 64 items. per page

Name	Type	State	Edit	Delete
About.com College Life	RSS	Published	Edit	Delete
Admin Welcome	Image	Published	Edit	Delete
Attribute Swapper	Portlet	Published	Edit	Delete
Audit DLM Fragments	Portlet	Published	Edit	Delete
Bookmarks	Bookmarks Portlet	Published	Edit	Delete
Cache Manager	Portlet	Published	Edit	Delete
Calendar	Portlet	Published	Edit	Delete
Courses	CMS	Published	Edit	Delete
Daily Business Cartoon	Image	Published	Edit	Delete
Demo Welcome	Image	Published	Edit	Delete

Select Portlet Type

Register New Portlet

Select Type

Option	Type	Description
<input type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input type="radio"/>	CMS	Displays configured HTML content after performing validation.
<input type="radio"/>	Portlet	Adapter for JSR-168 Portlets
<input type="radio"/>	Image	Displays an image with optional caption and subcaption
<input type="radio"/>	Inline Frame	Renders an HTML page within a frame. Not supported in browsers older than Internet Explorer 5 and Netscape 6.
<input type="radio"/>	RSS	News Feed
<input type="radio"/>	SQL	Executes a SQL query against a selected DataSource.
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet

Next

Cancel

Set basic portlet info

Register New Portlet

Summary Information

Option	Setting
Portlet Title:	<input type="text" value="Andrew Blog"/>
Portlet Name:	<input type="text" value="Andrew Blog"/>
Portlet Functional Name:	<input type="text" value="apetro_blog"/>
Portlet Description:	<input type="text" value="Andrew's Unicon blog."/>
Portlet Timeout:	<input type="text" value="5000"/> ms

Controls

Portlet Controls

- hasHelp
- editable
- hasAbout

Back

Next

Cancel

Portlet Parameters

Register New Portlet

portlet.xml Preferences

Default preferences provided by the portlets descriptor

Preference	Values	User Editable
------------	--------	---------------

RSS Feed

Enter the image URI, dimensions, border, and URL to link to if the image is clicked

Parameter	Value
-----------	-------

Parameter	Value	User Editable
Feed Name:	<input type="text" value="Andrew Unicon blog"/>	<input type="checkbox"/>
RSS Feed URL:	<input type="text" value="http://www.unicon.net/blo"/>	<input type="checkbox"/>
Summary View:	<input type="text" value="full"/>	<input type="checkbox"/>
Max Feed Entries:	<input type="text" value="5"/>	<input type="checkbox"/>

Select Categories

Select Categories

Which categories Andrew Blog belongs to

Groups:

All categories 

Please enter a name

Go

All categories Includes:

Categories

-  [Services](#)
-  [Finances](#)
-  [Research](#)
-  [Entertainment](#)
-  [Work](#)
-  [Development](#)
-  [News](#)
-  [Testing](#)
-  [Demonstration](#)
-  [Library](#)
-  [Advisor](#)
-  [Administration](#)
-  [Academics](#)
-  [Instructors](#)
-  [Information](#)
-  [uPortal](#)

Your Selections

 News


Select Users and Groups

PORTLET ADMINISTRATION [Return to dashboard](#) [Add to my layout](#)

Select People and Groups







Who can add Andrew Blog to their layout

Groups:


Everyone 

Everyone Includes:

People Groups

-  [Faculty](#)
-  [Staff](#)
-  [Portal System](#)
-  [Students](#)
-  [Guests](#)
-  [PAGS Root](#)

Your Selections

-  Admin User

What's a flow anyway?

What's a flow?

- A flow is a multi-step experience
- More than one request in making up a logical whole

A flow - what

- Has a beginning (state) and an end (state)
- Has zero or more intermediary states
- Has transitions between these states
- Has its own scope
 - Can conveniently remember state between these steps until the flow completes and your application is ready to process the whole result

Flows - why

- Re-use multi-step processes
- Revisit steps in flow
 - Constrained by allowable state transitions
 - Stateful!
 - Flow-scoped state (might) make your application cleaner (not having to model and remember at the persistence/domain layer incomplete state)


Re-use subflows

PORTLET ADMINISTRATION [Return to dashboard](#) [Add to my layout](#)

Select People and Groups







Who can add Andrew Blog to their layout

Groups:


Everyone 

Everyone Includes:

People Groups

-  [Faculty](#)
-  [Staff](#)
-  [Portal System](#)
-  [Students](#)
-  [Guests](#)
-  [PAGS Root](#)

Your Selections

-  Admin User

Revisit steps in flow

Back

Next

Cancel

Jasig News

Configuration

[Edit Configuration](#)

Configuration	Value
Portlet Title:	Jasig News
Portlet Name:	Jasig News
Portlet Functional Name:	jasig-news
Portlet Description:	Jasig News channel
Portlet Timeout:	10000
Portlet Type:	RSS
Edit:	true
Help:	true
About:	false

Parameters

[Edit Parameters](#)

Configuration	Value
Display Icon URL:	/ResourceServingWebapp/rs/tango/0.8.90/32x32/apps/internet-news-reader.png
Display Mobile Icon URL:	/uPortal/media/skins/icons/mobile/feed.png

portlet.xml Preferences

Preferences	Values	User Editable
summaryView	titleAndAbstract	true
maxStories	±0	true
newWindow	false	true
className	org.jasig.portlet.newsreader.adapter.RomeAdapter	true

Preferences

[Edit Preferences](#)

Preferences	Values	User Editable
summaryView	flyout	true
maxStories		true
name	Jasig News	true
url	http://jasig.org/jasig-news/feed	true

Categories

[Edit Categories](#)

- [News](#)

Groups

[Edit Groups](#)

- [Everyone](#)

Lifecycle Management

[Edit Lifecycle](#)

Option	Value
State	Published

[Save](#)[Cancel](#)[Edit Categories](#)[Edit Groups](#)[Edit Lifecycle](#)

States and Transitions

Views, Decisions, Actions, and Subflows

View States

```
<view-state id="viewLoginForm" />
```

- Renders a view.
- By convention, a view with the same name as the id. So, "viewLoginForm.jsp", e.g.

Transitions from a View

```
<view-state id="someView">  
  <transition on="submit" to="handleSubmit"/>  
  <transition on="cancel" to="mainMenu" />  
</view-state>
```

```
<input type="submit" name="_eventId_submit"  
value="Submit"/>  
<input type="submit" name="_eventId_cancel"  
value="Cancel"/>
```

```
<a href="{flowExecutionUrl}&_eventId=cancel">Cancel</a>
```

Decision States

```
<decision-state id="warn">  
  <if test="flowScope.warnCookieValue"  
then="showWarningView" else="redirect" />  
</decision-state>
```

- Boolean decision about what state is next

Action States

```
<action-state id="generateServiceTicket">  
  <evaluate expression="generateServiceTicketAction"/>  
  <transition on="success" to="warn"/>  
  <transition on="error" to="viewLoginForm"/>  
</action-state>
```

Subflow States

```
<subflow-state id="chooseGroup" subflow="entity-  
selector">  
  <input name="selectedGroups"  
value="portlet.groups"/>  
  <output name="selectedGroups"  
value="portlet.groups"/>  
  
  <transition on="back" to="chooseCategory"/>  
  <transition on="finish"  
to="chooseGroupNextScreen"/>  
</subflow-state>
```

How it really works

- Start at a start-state
- Flow through zero or more action and decision states until you get to another view state. Render it.
- Repeat previous step.
- Eventually exit the flow.

Ending with a view

```
<end-state id="viewServiceErrorView" />
```


Ending with a redirect

```
<end-state id="redirectView"  
  view="externalRedirect:${requestScope.response.url}"  
>
```

Ending with return from subflow

```
<end-state id="finish">  
    <output name="selectedGroups"  
value="flowScope.groups"/>  
    </end-state>
```

Setting variables and invoking Java

between and within states

Expression language

```
<evaluate
```

```
expression="authenticationViaFormAction.submit(flowRequestContext, flowScope.credentials, messageContext)" />
```

Storing result from expression

```
<evaluate  
expression="flowScope.service.getResponse(requestScope.  
serviceTicketId)" result-  
type="org.jasig.cas.authentication.principal.Response"  
result="requestScope.response" />
```

Where can I insert expressions?

- On flow start
- On state entry
- On view render
- On transition execution
- On state exit
- On flow end

Forms

Form Objects

- Form objects are POJOs
- JavaBean properties

```
public String getFname() {  
    return fname;  
}  
  
public void setFname(String name) {  
    fname = name;  
}  
...
```


Binding forms to model

```
<view-state id="chooseType" model="portlet">  
  <transition on="next" to="afterChooseType" />  
  <transition on="cancel" to="cancelEditScreen"  
bind="false"/>  
</view-state>
```

Model

Form fields bind to JavaBean properties of the model Java object.

Binding

- All properties bind by default
- Bindings can be explicitly declared
- You can also use a custom binder

Binding forms selectively to model

```
<view-state id="lifecycle" model="portlet">
  <binder>
    <binding property="lifecycleState"/>
    <binding property="publishHour"/>
    ...
  </binder>
  ...
</view-state>
```

Binder

Only bind the properties you intend to bind!

Selectively bind on transitions

```
<view-state id="viewLoginForm" view="casLoginView"
model="credentials">
  <binder>
    <binding property="username" />
    <binding property="password" />
  </binder>
  <transition on="submit" bind="true" to="handleSubmit"/>
  <transition on="cancel" bind="false" to="whyAreYouHere"
/>
</view-state>
```

Bind attribute on transition

You might not bother binding on a cancel, or you might bind but not validate if partial progress completing model.

Select Portlet Type

Register New Portlet

Select Type

Option	Type	Description
<input type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input type="radio"/>	CMS	Displays configured HTML content after performing validation.
<input type="radio"/>	Portlet	Adapter for JSR-168 Portlets
<input type="radio"/>	Image	Displays an image with optional caption and subcaption
<input type="radio"/>	Inline Frame	Renders an HTML page within a frame. Not supported in browsers older than Internet Explorer 5 and Netscape 6.
<input type="radio"/>	RSS	News Feed
<input type="radio"/>	SQL	Executes a SQL query against a selected DataSource.
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet


Next

Cancel

Next without required field

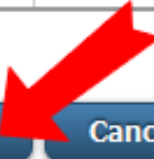
PORTLET ADMINISTRATION

Register New Portlet

 Please choose a portlet type

Select Type

Option	Type	Description
<input type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input type="radio"/>	SQL	Executes a SQL query against a selected DataSource
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet



But cancel doesn't require field

Register New Portlet

Select Type

Option	Type	Description
<input type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input type="radio"/>	CMS	Displays configured HTML content after performing validation.
<input type="radio"/>	Portlet	Adapter for JSR-168 Portlets
<input type="radio"/>	Image	Displays an image with optional caption and subcaption
<input type="radio"/>	Inline Frame	Renders an HTML page within a frame. Not supported in browsers older than Internet Explorer 5 and Netscape 6.
<input type="radio"/>	RSS	News Feed
<input type="radio"/>	SQL	Executes a SQL query against a selected DataSource.
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet

Next

Cancel

Binding forms to model

```
<view-state id="chooseType" model="portlet">  
  <transition on="next" to="afterChooseType" />  
  <transition on="cancel" to="cancelEditScreen"  
bind="false"/>  
</view-state>
```

Model

Form fields bind to JavaBean properties of the model Java object.

Properties can be required

```
<view-state id="viewLoginForm" view="casLoginView"  
model="credentials">  
  <binder>  
    <binding property="username" required="true" />  
    <binding property="password" required="true" />  
  </binder>  
  
</view-state>
```

CAS login form

Enter your NetID and Password

NetID:

Password:

Warn me before logging me into other sites.

LOGIN

clear

Required fields



Username is a required field.



Password is a required field.

Enter your NetID and Password

NetID:

Password:

Warn me before logging me into other sites.

LOGIN

clear

Enter your NetID and Password

NetID:

Password:

Warn me before logging me into other sites.

LOGIN

clear

Properties can be required

```
<view-state id="viewLoginForm" view="casLoginView"  
model="credentials">  
  <binder>  
    <binding property="username" required="true" />  
    <binding property="password" required="true" />  
  </binder>  
  
</view-state>
```

Validation

- Validator for model objects
- Or Validator for model-in-specific-state
- As in, you can program custom validators in Java

Selectively validate on transitions

```
<view-state id="basicInfo" model="portlet">  
  <transition on="back" to="chooseType"  
validate="false"/>  
  <transition on="next" to="setParameters"/>  
</view-state>
```

validate attribute on transition

You might bind but not validate if you want to capture the user input but not prevent transition if the input is invalid.

Say I pick a portlet type

Select Type

Option	Type	Description
<input checked="" type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input type="radio"/>	CMS	Displays configured HTML content after performing validation.
<input type="radio"/>	Portlet	Adapter for JSR-168 Portlets
<input type="radio"/>	Image	Displays an image with optional caption and subcaption
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet

Next

Cancel


I fill out this form, but...

PORTLET ADMINISTRATION

Register New Portlet

Summary Information

Option	Setting
Portlet Title:	<input type="text" value="Not so advanced CMS"/>
Portlet Name:	<input type="text" value="I changed my mind"/>
Portlet Functional Name:	<input type="text"/>
Portlet Description:	<input type="text" value="Actually, I want just 'CMS'"/>
Portlet Timeout:	<input type="text" value="5000"/> ms



So I pick another portlet type

PORTLET ADMINISTRATION ← Return

Register New Portlet

Select Type

Option	Type	Description
<input type="radio"/>	Advanced CMS	Displays configured HTML content.
<input type="radio"/>	Bookmarks Portlet	Bookmarks Portlet
<input checked="" type="radio"/>	CMS	Displays configured HTML content after performing validation.
<input type="radio"/>	Portlet	Adapter for JSR-168 Portlets
<input type="radio"/>	Image	Displays an image with optional caption and subcaption
<input type="radio"/>	Web Proxy Portlet	Web Proxy Portlet

And my form data is right there.

Summary Information

Option	Setting
Portlet Title:	<input type="text" value="Not so advanced CMS"/>
Portlet Name:	<input type="text" value="I changed my mind"/>
Portlet Functional Name:	<input type="text"/>
Portlet Description:	<input type="text" value="Actually, I want just 'CMS'"/>
Portlet Timeout:	<input type="text" value="5000"/> ms

Even though it's not valid

ⓘ Please enter a fname

Summary Information

Option	Setting
Portlet Title:	Not so advanced CMS
Portlet Name:	I changed my mind
Portlet Functional Name:	
Portlet Description:	Actually, I want just "CMS"
Portlet Timeout:	5000 ms

Controls

Portlet Controls

- hasHelp
- editable
- hasAbout

Back

Next

Cancel

Selectively validate on transitions

```
<view-state id="basicInfo" model="portlet">  
  <transition on="back" to="chooseType"  
  validate="false"/>  
  <transition on="next" to="setParameters"/>  
</view-state>
```

validate attribute on transition

You might bind but not validate if you want to capture the user input but not prevent transition if the input is invalid.

PortletDefinitionForm

```
public class PortletDefinitionForm implements
Serializable {

    public String getFname() {
        return fname;
    }

    public void setFname(String name) {
        fname = name;
    }

    ...
}
```

PortletDefinitionFormValidator

```
public void validateBasicInfo (PortletDefinitionForm
def,          MessageContext ctx) {
    if (StringUtils.isEmpty(def.getFname())) {
        ctx.addMessage (new
MessageBuilder().error().source ("fName")
            .code ("fname.required").build());
    } else if (!
FunctionalNameType.isValid(def.getFname())) {
        ctx.addMessage (new
MessageBuilder().error().source ("fName")
            .code ("fname.invalid").build());
    }
}
...

```

Validate\${viewState}()

```
public void validateBasicInfo (PortletDefinitionForm
def,          MessageContext ctx) {
    . . .
}
```

```
<view-state id="basicInfo" model="portlet">
    <transition on="back" to="chooseType"
validate="false"/>
    <transition on="next" to="setParameters"/>
</view-state>
```

Scopes

Scopes

- Flash scope
- View Scope
- Flow Scope
- Conversation Scope

Flash scope

- Cleared when a view is rendered

View Scope

- Created when enter view-state
- Destroyed on exit view-state
- Useful for transient state for rendering the view
 - That you might need on processing the event fired by the user interaction with the view
 - Or that you need for re-rendering the view

Flow Scope

- Created when enter flow
- Destroyed on exit flow
- Useful for transient state within the flow

Conversation scope

- Persists across returns from subflows

Conclusions

Spring Web Flow in uPortal 4

- Many administrative portlets in uPortal 4 implemented using Spring Web Flow

PORTLET ADMINISTRATION [Return to dashboard](#) [Add to my layout](#)

Select People and Groups

Who can add Andrew Blog to their layout

Groups:

Everyone

Everyone Includes:

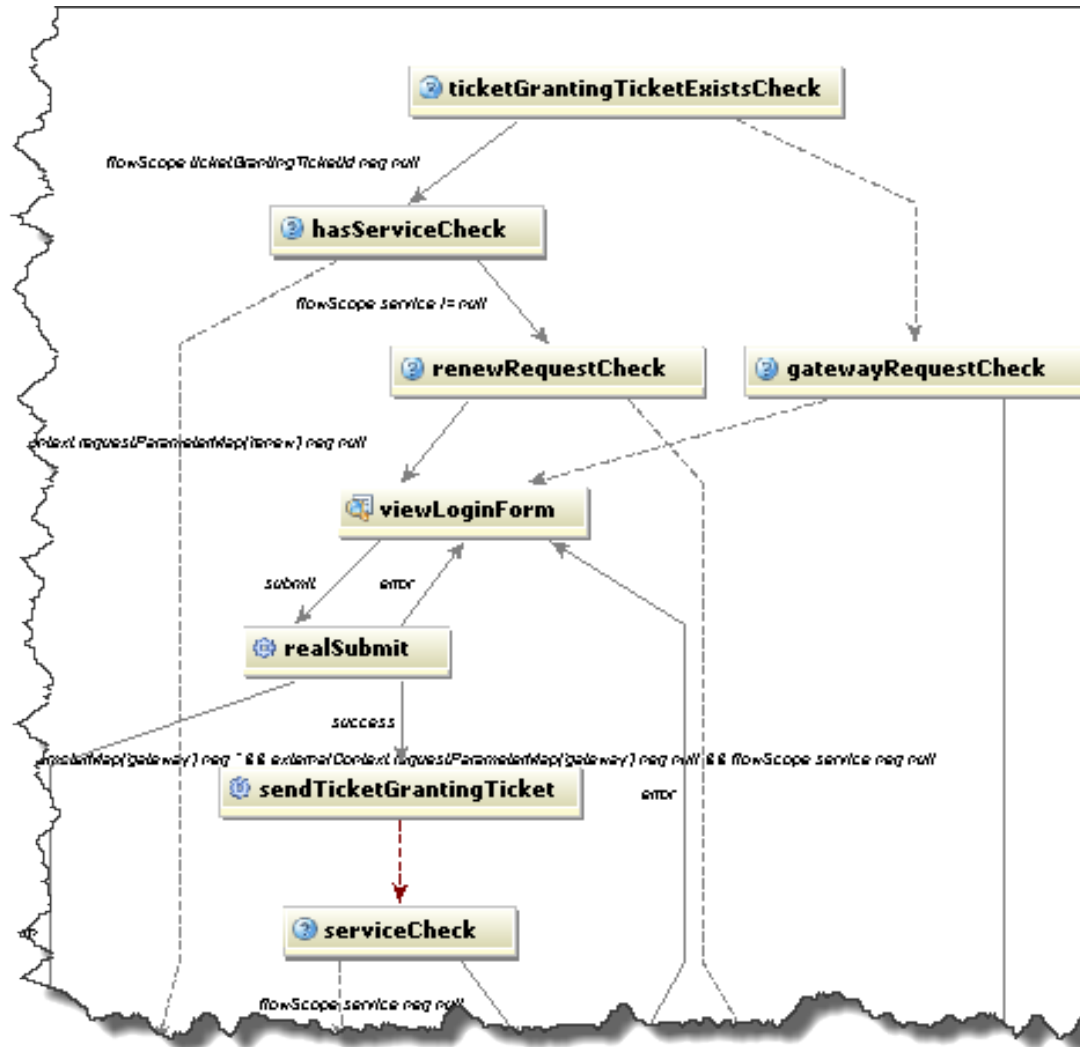
People Groups

- Faculty
- Staff
- Portal System
- Students
- Guests
- PAGS Root

Your Selections

- ✓ Admin User

Spring Web Flow in CAS



Spring Web Flow for Portlet Dev

- Works with Spring PortletMVC

Spring Web Flow Web Development

- Quickly develop self-service web flows
- Update, maintain, and tweak these applications with panache

Contact Information



Andrew Petro
Software Developer
Unicon, Inc.

apetro@unicon.net
www.unicon.net/blog/apetro